SPRAWOZDANIE

Zajęcia: Nauka o danych II

Prowadzący: prof. dr hab. Vasyl Martsenyuk

Laboratorium Nr 2 Data 10.05.2025 Temat: „Projektowanie zaawansowanych architektur sieci neuronowych w TensorFlow lub PyTorch” Wariant 2	Anna Więzik Informatyka II stopień, niestacjonarne, 2 semestr, gr.1a TTO

1. Polecenie:

Link do repozytorium: https://github.com/AnaShiro/NoD2_2025

Zadanie 2 (U-Net): Segmentuj komórki w obrazach mikroskopowych — BBBC038.
Dane: Obrazy grayscale, przygotuj maski binarne. Zastosuj augmentację (obrót, odbicie).

Nowoczesne architektury sieci neuronowych, takie jak U-Net, Autoencoder, Encoder-Decoder, BRNN, GAN i Transformer, znacząco zwiększyły możliwości głębokiego uczenia. U-Net doskonale sprawdza się w segmentacji obrazów dzięki symetrycznej strukturze i połączeniom skip. Autoenkodery pozwalają na kompresję oraz rekonstrukcję danych. Modele encoder-decoder i BRNN efektywnie przetwarzają sekwencje, uwzględniając kontekst. GAN-y generują realistyczne dane, a Transformery, bazujące na mechanizmie uwagi, zapewniają wysoką skuteczność i umożliwiają równoczesne przetwarzanie sekwencji.

2. Opis programu opracowanego

```python
import tensorflow as tf
from tensorflow.keras import layers, models
import matplotlib.pyplot as plt

input_dim = 32 * 32 * 3
encoding_dim = 256

input_img = tf.keras.Input(shape=(input_dim,))
encoded = layers.Dense(512, activation='relu')(input_img)
encoded = layers.Dense(encoding_dim, activation='relu')(encoded)

decoded = layers.Dense(512, activation='relu')(encoded)
decoded = layers.Dense(input_dim, activation='sigmoid')(decoded)

autoencoder = models.Model(input_img, decoded)
autoencoder.compile(optimizer='adam', loss='mse')

autoencoder.summary()

(x_train, _), (x_test, _) = tf.keras.datasets.cifar10.load_data()
x_train = x_train.reshape(len(x_train), input_dim) / 255.0
x_test = x_test.reshape(len(x_test), input_dim) / 255.0

history = autoencoder.fit(x_train, x_train, epochs=10, batch_size=256, validation_data=(x_test, x_test), verbose=0)

plt.plot(history.history["loss"], label="Train Loss")
plt.plot(history.history["val_loss"], label="Val Loss")
plt.xlabel("Epoka")
plt.ylabel("Strata rekonstrukcji")
plt.legend()
plt.grid()
plt.show()
```

✓ 29.8s

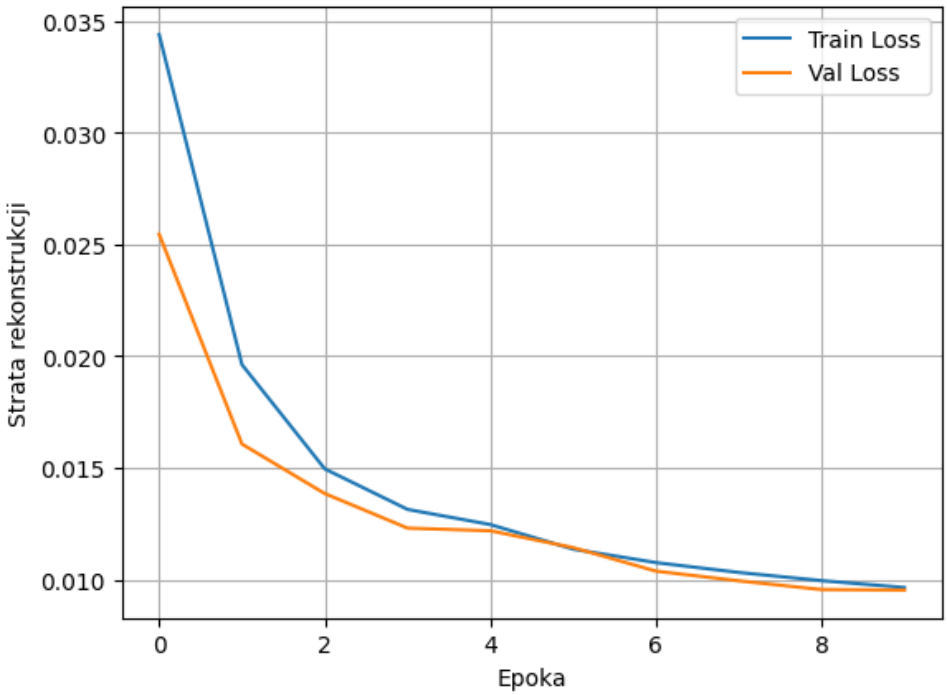
Model: "functional"

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 3072)	0
dense (Dense)	(None, 512)	1,573,376
dense_1 (Dense)	(None, 256)	131,328
dense_2 (Dense)	(None, 512)	131,584
dense_3 (Dense)	(None, 3072)	1,575,936

Total params: 3,412,224 (13.02 MB)

Trainable params: 3,412,224 (13.02 MB)

Non-trainable params: 0 (0.00 B)



3. Wnioski

Dzięki ćwiczeniu uczestnicy zdobyli praktyczne doświadczenie w pracy z zaawansowanymi architekturami sieci neuronowych i ich zastosowaniami. Zrozumieli, jak wybór modelu wpływa na efektywność w zadaniach takich jak segmentacja, przetwarzanie sekwencji czy generowanie danych. Zdobyta wiedza stanowi solidną bazę do samodzielnego projektowania nowoczesnych rozwiązań w zakresie głębokiego uczenia.